ดร. ปัญญา เปรมปรีดิ์

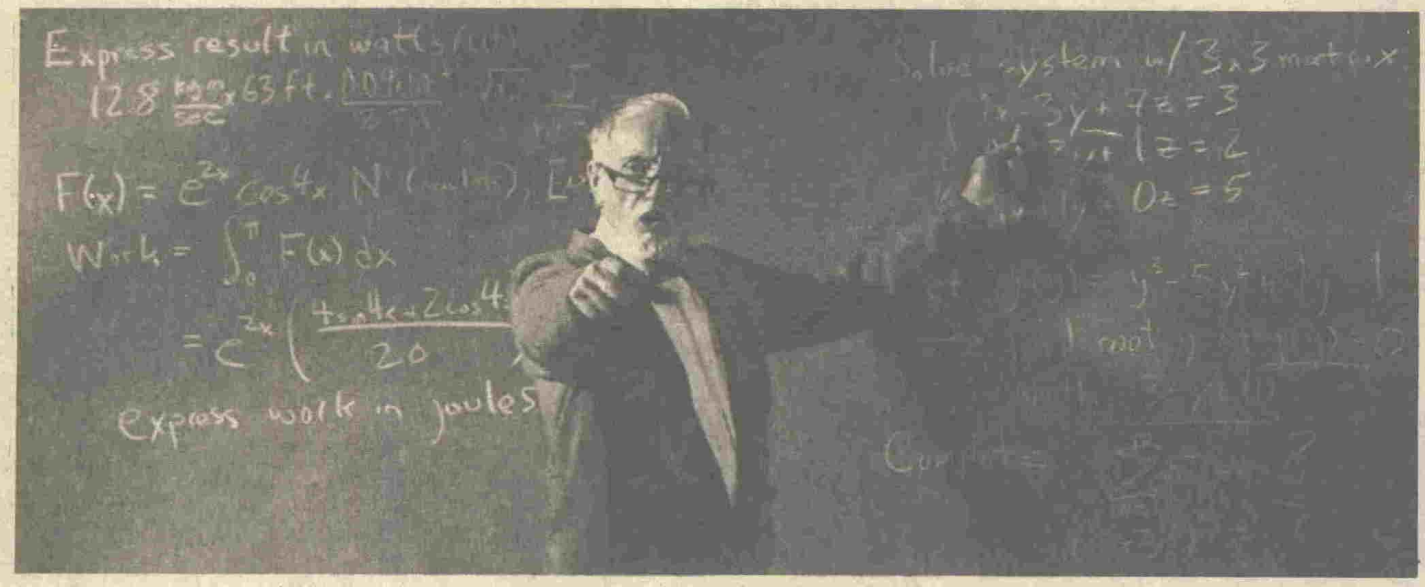
ทำไมคนจึงอยากจะเป็นแต่ Analyst

ทำไมเขาจึงไม่อยาก
เป็นโปรแกรมเมอร์
ทัศนคติที่น่าเศร้า
ในวงการคอมพิวเตอร์

เมื่อ 20 ปีก่อนโน้นผู้เขียนเคยเจอกับปัญหาอันหนึ่งคือ คนที่เข้ามาร่วมโครงการการใช้คอมพิวเตอร์ของธนาคารแห่งหนึ่งต่างแย่งชิงกันขอบรรจุเข้าเป็น "อะนารีสต์" (Analyst) ธนาคารแห่งนั้นสั่งซื้อเครื่องเมนเฟรมราคาประมาณ 16 ล้านบาท และดึงคนจากฝ่ายงานต่าง ๆ ประมาณ 40 คนเข้าฝึกอบรมการเขียนโปรแกรม การวิเคราะห์ระบบ และการออกแบบระบบ คนเหล่านี้จะถูกแบ่งออกเป็น 2 กลุ่ม คือเป็น "โปรแกรมเมอร์" กับเป็น "อะนารีสต์" ในตอนนั้นเขาแบ่งแยกงานออกเป็น 2 ส่วนอย่างชัดเจน ทั้งนี้โดยทำตามคำแนะนำในตำราฝรั่ง ผู้เขียนเองนั้นมีหน้าที่ไปช่วยฝึกอบรม ถึงแม้ว่าจะมีความรู้สึกไม่ค่อยเห็นด้วย แต่ก็ไม่ได้ท้วงติง ผลการใช้แนวคิด "แยกงานเป็น 2 ลักษณะอย่างชัดเจน" นั้นเป็นอย่างไรผมคงไม่ต้องพูดถึง เพราะในที่สุดธนาคารแห่งนั้นก็ได้จับงาน 2 หน้าที่นี้รวมเข้าด้วยกัน แล้วให้มีพนักงานเพียงกลุ่มเดียวซึ่งเรียกว่า "โปรแกรมเมอร์/อะนารีสต์

เหตุผลและทัศนคติ

นั่นเป็นเหตุการณ์เมื่อ 15-20 ปีมาแล้ว แต่ปัญหานี้ดูจะยังไม่หมดไป ทุกครั้งที่ผมต้องไปนั่งเป็นกรรมการสอบสัมภาษณ์คนเข้าทำงาน โดยเฉพาะอย่างยิ่งคือจากกลุ่มนักเรียนนอก พบว่าพวกเขาส่วนใหญ่ยังอยากเป็นแต่อะนารีสต์ เรื่องนี้ทำให้รู้สึกอึดอัดใจมาโดยตลอด แต่ในที่สุดผมก็พบกับความจริงคือเดิมนั้นผมรู้สึกรำคาญอย่างยิ่งที่มีคนพยายามจะเลือกเป็นแต่อะนารีสต์ โดยไปเข้าใจว่าพวกเขาหัวสูง เหยียบขี้ไก่ไม่ฝ่อ หรือเห่อเหิมมากไปหน่อย พวกจบจากนอกนั้นยิ่งมีแนวโน้มที่จะยกตนให้สูงอยู่แล้ว ผมจึงเกลียดมาก แต่หลังจากที่ได้ซักถามว่า "มีเหตุผลอะไรที่ไม่ขอเป็นโปรแกรมเมอร์" ก็ได้คำตอบคือ เด็ก ๆ เหล่านี้ให้คำจำกัดความของคำว่า "อะนารีสต์" และ "โปรแกรมเมอร์" ไม่ตรงกับเรา และต้นเหตุใหญ่มาจากการคาดเดาอย่างผิด ๆ ของพวกเขา พวกเขาไม่มีประสบการณ์ พวกเขาไม่รู้เลยว่าเราปฏิบัติงานกันอย่างไร และมหาวิทยาลัยต่าง ๆ นั้นต่างก็ให้ทัศนคติที่ผิด ๆ แก่พวกเขา เรื่องนี้จึงกลายเป็นเรื่องเศร้าสำหรับหลาย ๆ คน ทั้งนี้เพราะผมจะไม่รับใครที่ยังมีความคิดเห็นที่ขัดแย้งกับผมเข้ามาทำงาน และเชื่อว่ายังมีหัวหน้าศูนย์อีกหลายคนที่ทำอย่างเดียวกับผม เรื่องนี้เกิดขึ้นได้



อย่างไร และพวกเรากับพวกเขาเข้าใจผิดกันที่ตรงไหน ผมจะขอขยายความให้ฟัง

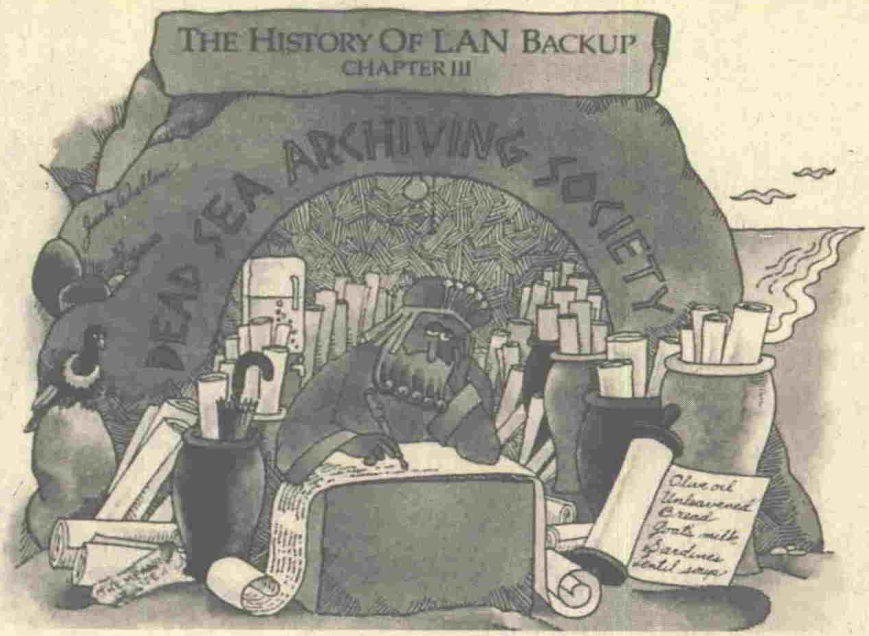
สู่โลกความจริง

พวกเราที่เคยทำงานด้านพัฒนาระบบงานคอมพิวเตอร์มาแล้วนั้น คงจะเห็นพ้องกับผมว่า งานวิเคราะห์และออกแบบระบบนั้นใช้ช่วงเวลาที่สั้นมาก บางคนอาจบอกไม่ได้ด้วยซ้ำไปว่าไอเดียมันเกิดขึ้นมาได้อย่างไร นักออกแบบระบบนั้นเมื่อได้รับมอบหมายงาน ก็เกือบจะบอกได้อยู่แล้วว่าระบบนั้นจะต้องมีแฟ้มข้อมูลอะไรบ้าง เราต้องแวะไปคุยกับเจ้าของงานเพื่อเก็บรายละเอียดเรื่องขนาดความยาวของฟิลด์ต่าง ๆ การแยกประเภทหรือรหัสต่าง ๆ และเช็คให้แน่ใจว่าขอบข่ายของงานเป็นดังที่เราวาดไว้หรือไม่ ในบางครั้งเราก็อาจต้องวางโครงสร้างฐานข้อมูลใหม่ ทั้งนี้เพราะขอบเขตของงานกว้างกว่าที่เราคิดไว้ แต่ส่วนใหญ่แล้วเค้าโครงของแฟ้มข้อมูลหรือฐานข้อมูลจะเป็นดังที่เราคิดไว้ สาเหตุที่เป็นเช่นนี้ก็เพราะงานดาต้าโปรเซสซิงนั้นมันเหมือน ๆ กันหมด ทุกระบบก็ต้องมีมาสเตอร์ไฟล์, ต้องมีตารางรหัสข้อมูล, มีแฟ้มรายการที่นำมาอัพเดต, ต้องมีไฟล์ชั่วคราวเพื่อการออกรายงาน ฯลฯ สิ่งที่แตกต่างกันคือ จำนวนฟิลด์ ขนาดของฟิลด์ และขนาดของไฟล์ สิ่งที่เราจะเล่นพลิกแพลงได้ก็จะเป็นเรื่องของการเข้าถึงข้อมูล (Access Method) ซึ่งปัจจุบันก็เกือบจะถูกกำหนดตายตัวไปแล้ว ทั้งนี้เพราะงานส่วนใหญ่เปลี่ยนมาเป็นแบบออนไลน์-เรียลไทม์ การเข้าถึงข้อมูลก็จะเป็นแบบอินเด็กซ์ หรืออินเด็กซ์ซีเควนเชียล (Indexed, Indexed Sequential) เท่านั้น

ครับ นั่นคืองานที่ต้องใช้ความคิดและท้าทายที่สุด แต่มันมีช่วงเวลาที่สั้นมาก ต่อจากนั้นเราก็ต้องมากำหนดว่าต้องมีโปรแกรมกี่โปรแกรม ชื่ออะไรบ้าง และแต่ละโปรแกรมให้ทำหน้าที่อะไร เรื่องนี้ก็เช่นกัน คือเป็นงานที่เข้าลักษณะเป็นรูทีนไปแล้ว คือต้องมีโปรแกรมเพื่อสร้างฐานข้อมูล (Create) ต้องมีโปรแกรมเพื่ออัพเดตข้อมูล (Update) ต้องมีโปรแกรมดึงข้อมูลออกมา (Extract) แล้วก็ต้องมีโปรแกรมทำรายงาน (Reporting) การทำรายงานในสมัยใหม่นี้เราก็มุ่งเน้นให้ออกที่จอภาพมากขึ้น ซึ่งเรามักจะเรียกว่า โปรแกรมประเภทสอบถาม (Enquiry) สำหรับคนที่คิดไกลไปอีกหน่อยก็จะคิดเผื่อคนทางสายงานตรวจสอบด้วย คือมีโปรแกรมประเภทเพื่อการควบคุมและสอบยัน (Audit & Control) และคนที่เคยรู้ฤทธิ์ของ "บัก" ก็จะทำโปรแกรมประเภทช่วยดูแลและแก้ไขแฟ้มข้อมูลยามฉุกเฉิน (Special Maintenance)

ทำความเข้าใจใหม่

ทีนี้ก็จะมาถึงจุดที่เราเข้าใจผิดซึ่งกันและกัน พวกเราที่มีประสบการณ์ในการพัฒนาระบบแล้วนั้นจะบอกว่างานออกแบบระบบจะมาหยุดที่ตรงจุดนี้ คือ ผู้ออกแบบระบบจะเขียนหรืออธิบายอย่างสั้น ๆ ว่าแต่ละโปรแกรมทำหน้าที่อะไร หรือคือกำหนดฟังก์ชันของโปรแกรม (Program Function) เท่านั้น เรา



ไม่ได้ลงลึกไปจนถึงขนาดบอกว่าต้องเขียนโปรแกรมเป็นขั้นตอนอย่างไร การมอบหมายงานจะต้องพูดให้ละเอียดหรือไม่ก็จะอยู่กับตัวผู้รับหน้าที่เขียนโปรแกรม คือถ้าเป็นเด็กเพิ่งจบมาใหม่ ๆ หรือคนที่เพิ่งเข้ามาร่วมงานใหม่ ๆ เราก็จะอธิบายละเอียดหน่อย ทั้งนี้เพื่อให้ได้โปรแกรมที่มีโครงสร้างและมีรูปประโยคที่เข้ามาตรฐานของศูนย์ แต่ถ้าเป็นคนเก่าคนแก่ที่เคยทำงานมาด้วยกันหลายครั้งแล้วเราจะปล่อยให้เป็นหน้าที่ของเขาที่จะไปวางโครงสร้างของโปรแกรม (Program Flowchart) เองเลย

การวางโครงสร้างของโปรแกรมเป็นงานที่ท้าทายอีกชิ้นหนึ่ง และเป็นงานที่พวกนักเรียนที่เพิ่งจบมาใหม่ ๆ อยากทำ แต่พวกเขาคิดว่านี่เป็นหน้าที่ของอะนารีสต์ คือเขาไปคิดว่าอะนารีสต์ต้องลงไปทำรายละเอียดเรื่องโครงสร้างของโปรแกรม ตัวที่ทำให้เขาเข้าใจผิดก็คือ ในมหาวิทยาลัยจะสอนเรื่องการใช้คำสั่งระดับกลาง (Pseudo Code) และเรื่องการเขียนโปรแกรมอย่างมีโครงสร้าง (Structured Programming) สองวิชานี้ทำให้พวกเขาต้องมาเขียนรายละเอียดของโปรแกรม นอกจากนี้แล้วพวกเขายังคิดว่า ต้องเขียนบรรยายถึงขั้นตอนการทำงานของโปรแกรมอย่างละเอียด เพราะต้องส่งมอบให้กับโปรแกรมเมอร์ที่เขาไม่รู้จักหน้าค่าตา พวกเขาไปคิดวาดภาพเหมือนกับการทำงานในองค์กรขนาดใหญ่อย่างนาซ่า (NASA) ซึ่งมีอะนารีสต์ทำงานอยู่เมืองหนึ่ง แต่มีโปรแกรมเมอร์ (มือปืนรับจ้าง) อยู่กระจายทั่วประเทศ

ครับ นี่คือจุดที่เราเข้าใจผิดกันอย่างแรงและแก้ไขยากด้วย ทั้งนี้เพราะเรายังมีนักคอมพิวเตอร์บางคน กับผู้บริหารบางประเภทที่พยายามจะพูดว่าอะนารีสต์ต้องเขียนรายละเอียดของโปรแกรม และส่งมอบให้โปรแกรมเมอร์ไปทำโปรแกรม คนประเภทนี้เป็นนักวิชาการ เขาคิดว่า "รายละเอียดของโปรแกรม" คือข้อกำหนด (Spec) หรือข้อตกลงระหว่างคน 2 คน เขาคิดว่าสิ่งนี้ต้องขีดเขียนเป็นลายลักษณ์อักษรเอาไว้ ทั้งนี้เพื่อจะได้เป็นตัวอ้างอิง เขาคิดว่าอะนารีสต์ต้องเป็นผู้เขียน และโปรแกรมเมอร์ต้องทำตามโดยเคร่งครัด บางคนคิดเลยเถิดไปจนถึงขั้นว่า ถ้าเขียนข้อกำหนดของโปรแกรมไปผิด ๆ โปรแกรมเมอร์ก็มีสิทธิ์ที่จะเขียนโปรแกรมอย่างผิด ๆ ตามไปด้วย เขาไปวาดภาพเป็นการทำงานแบบจ้างเหมาเป็นทอด ๆ แต่ในชีวิตจริงนั้นส่วนใหญ่โปรแกรมเมอร์และอะนารีสต์อยู่ในศูนย์คอมพิวเตอร์เดียวกัน

ทำไมไม่อยากเป็นโปรแกรมเมอร์

ทำไมผมจึงจับความผิดปกติอันนี้ได้?

ก็มาจากคำถามว่า "ทำไมคุณจึงไม่อยากเป็นโปรแกรมเมอร์" ยังไงล่ะครับ เพราะคนที่มาสมัครงานเขาตอบเป็นเสียงเดียวกันว่า "โปรแกรมเมอร์ต้องเขียนโปรแกรมตามคำสั่งคนอื่น งานแบบนั้นไม่ท้าทาย ผมต้องการออกแบบลอจิกของโปรแกรมเอง"

ฟังคำตอบอันนี้แล้วท่านผู้อ่านคงจะต้องรู้สึก "จุกในอก" ทั้งนี้เพราะจะอธิบายให้เข้าใจกันนั้นต้องใช้เวลานานทีเดียว ผมได้พยายามดูแล้วคือได้พยายามที่จะบอกเขาไปว่าเราทำงานเป็นทีม เราให้ช่วยกันคิด คนที่มีอาวุโสจะเป็นผู้เสนอรูปแบบหรือโครงสร้างของระบบทุกคนมีสิทธิ์ติและชม หลังจากนั้นหัวหน้าทีมก็จะกำหนดชื่อและหน้าที่ของโปรแกรมต่าง ๆ แล้วแบ่งแยกโปรแกรมไปให้ลูกทีมช่วยกันเขียน หัวหน้าทีมก็อาจต้องรับไปเขียนด้วย ใครที่ยังไม่เข้าใจก็อาจซักถามจากหัวหน้าทีม ใครที่เขียนโปรแกรมไปแล้วติดปัญหาบางอย่าง เช่น เกิดความไม่สอดคล้องกันในหน้าที่ของโปรแกรมก็อาจนำมาหารือกัน แล้วแก้ไขข้อกำหนดหรือหน้าที่ของโปรแกรมเสียใหม่ ข้อกำหนดเหล่านี้เราร่างไว้สั้น ๆ ย่อ ๆ เมื่อพัฒนาระบบจนเสร็จแล้วจึงจะนำมาแก้ไขและพิมพ์ บางทีมอาจไม่มีเอกสารไว้เลย ทำเสร็จแล้วค่อยเขียนตามหลัง ฯลฯ เมื่อผม

อธิบายไปแบบนี้ก็พบว่าเขามีคำถามเยอะแยะ เช่น โปรแกรมเมอร์จะเขียนโปรแกรมได้อย่างไร จะเอาเอกสารอะไรมาอ้างอิง และถ้าคนลาออกจะทำอย่างไร เวลาทดสอบระบบแล้วพบว่าโปรแกรมของคนบางคนทำงานไม่สอดคล้องกับคนอื่นแล้วจะพิสูจน์ได้อย่างไรว่าใครผิดใครถูก และใครควรจะแก้ไขโปรแกรม

การอธิบายหรือตอบคำถามต่าง ๆ ข้างบนนี้ทำไม่ได้ในช่วงเวลาที่เราสอบสัมภาษณ์กันอยู่ เพราะถ้าจะมานั่งโต้ตอบกันอยู่ก็จะเกิดปัญหากับกรรมการสอบสัมภาษณ์ท่านอื่น ๆ เช่น กรรมการจากฝ่ายการพนักงานและประธานสอบจะไม่สามารถตั้งคำถามเพื่อประเมินทางด้านอื่น และท่านเหล่านั้นก็ไม่สนใจในรายละเอียดระหว่างหน้าที่ของอะนารีสต์กับโปรแกรมเมอร์ ท่านจะเห็นว่าเป็นการเสียเวลา ดังนั้นเรื่องนี้จึงเป็นกรรมของทั้งผู้สมัครสอบและฝ่ายคอมพิวเตอร์เราไม่กล้ารับเข้ามาร่วมทีมเพราะไม่แน่ใจว่าจะปรับความเข้าใจกันได้หรือไม่ เพราะบางทีผมก็พบว่าผมตัดสินใจถูกแน่ ๆ เพราะบางคนบอกเลยว่า "งานเขียนโปรแกรมเป็นงานจำเจ ถึงจะให้มีสิทธิ์คิดวางใจลอจิกเองผมก็ไม่เอา ผมออกแบบและควบคุมโปรเจ็กต์จะดีกว่า ผมสามารถทำประโยชน์ให้กับศูนย์ของท่านได้มากกว่า" คนแบบนี้ทำงานด้วยกันไม่ได้

ทำไมผมจึงรังเกียจคนประเภท "ขอเป็นอะนารีสต์ และทำด้านออกแบบระบบแต่เพียงอย่างเดียว" มากมายนัก

ชี้แจงความกระจ่าง

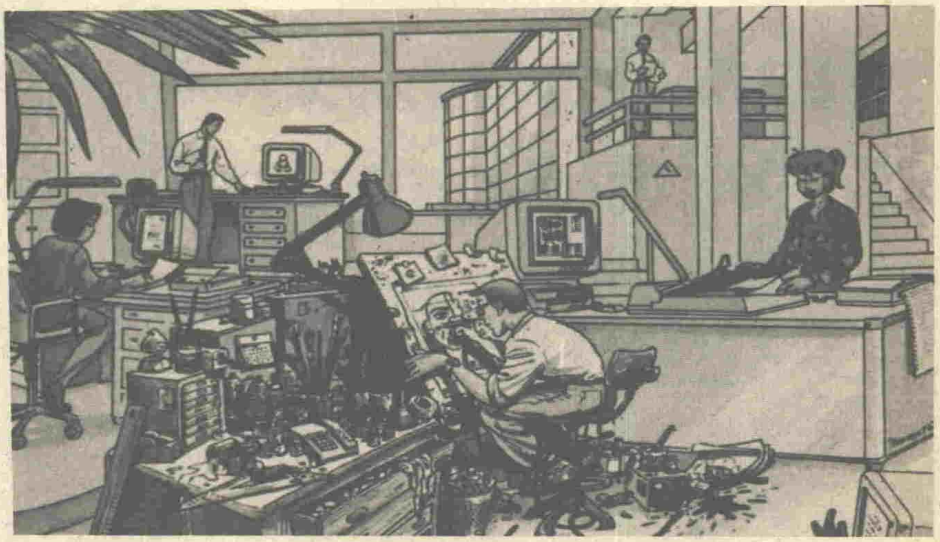
เรื่องนี้คงต้องอธิบายกันนานหน่อย ประการแรกก็คือ ถ้าคนหนึ่งขอเป็นผู้ออกแบบระบบแต่เพียงอย่างเดียว เราก็จะต้องมีคนที่ต้องทำหน้าที่เขียนโปรแกรมแต่เพียงอย่างเดียวอีกจำนวนหนึ่ง อัตราส่วนระหว่างจำนวนอะนารีสต์ต่อโปรแกรมเมอร์นี้กะคร่าว ๆ ก็จะเป็นประมาณ 1 ต่อ 5 ทั้งนี้เพราะการออกแบบระบบนั้นใช้เวลาและแรงงานประมาณ 1/5 ของการทำโปรแกรม จากอัตราส่วนดังกล่าวนี้เราก็จะมีคำถามว่าโปรแกรมเมอร์ 5 คนนั้นเราจะเอามาจากไหน คนเหล่านี้จะให้ผ่านการศึกษาประเภทใด และเราจะให้เขาทำหน้าที่เป็นโปรแกรมเมอร์ไปจนตายได้อย่างไร เรื่องเหล่านี้ผมคิดว่าคนที่ขอเป็นอะนารีสต์ไม่เคยได้คิด เพราะถ้าเขาลงนั่งคิดอย่างละเอียดถี่ถ้วนแล้วจะไม่มีหนทางเป็นไปได้ ทางแรกที่เขาอาจเสนอได้คือไปเอามาจากพวกจบ ปวช., ปวส. หรืออนุปริญญาต่าง ๆ คนระดับนั้นอาจยอมทำงานรูทีนตลอดชีวิตได้ (จำนวนหนึ่ง) แต่ในโลกแห่งความเป็นจริง เราก็ทราบกันดีว่าไม่มีหลักสูตร ปวช., ปวส. ที่สอนคนให้ออกมาทำงานเป็นโปรแกรมเมอร์และเชื่อด้วยว่าเจ้าของโรงเรียนเหล่านี้เขาได้คิดแล้ว เขารู้ว่าผลิตหลักสูตร "โปรแกรมเมอร์" ออกมาแล้วจะไม่สำเร็จ ทั้งนี้เพราะค่าใช้จ่ายสูงและโอกาสที่ลูกศิษย์ลูกหาจะออกไปทำมาหากินให้เจริญนั้นมีน้อย ทั้งนี้เพราะจริง ๆ แล้วคนที่จะเป็นโปรแกรมเมอร์ฝีมือดี ๆ นั้นต้องมีพื้นฐานสูง โปรแกรมเมอร์ที่เรามีกันอยู่ในปัจจุบันจบปริญญาตรีกันเกือบทั้งนั้น ขนาดจบปริญญาตรีมาเรายังทำโปรแกรมกันได้ไม่ค่อยดีนักแล้วเราจะคาดหวังอะไรจากคนระดับล่างลงไป เรามีช้างเผือกอยู่ไม่กี่คนหรอก ช้างเผือกในที่นี้ผมหมายถึง พวกที่เรียนแค่ระดับปวช., ปวส. แต่มีลอจิกดีมาก สามารถเขียนโปรแกรมได้เก่ง นอกจากปัญหานี้ เจ้าของโรงเรียนระดับ ปวช., ปวส. เขาก็ต้องคิดถึงคู่แข่งในตลาดด้วย ในตลาดเรามีโรงเรียนสอนเรื่องคอมพิวเตอร์แบบระยะสั้นเปิดกันอยู่เกลื่อนกลาด พวกนี้คิดค่าเรียนเพียงคอร์สละ 1,000-2,000 บาท ผลิตลูกศิษย์ลูกหาออกมาเป็นแสนคนแล้ว แต่ท่านที่มีประสบการณ์ก็คงบอกได้ว่าผลผลิตเหล่านี้เป็นอย่างไร เรายังไม่สามารถแก้ปัญหา "ขาดโปรแกรมเมอร์" ได้ เพราะคนเรียนคอร์สเหล่านี้แล้วก็ยังไม่สามารถไปปฏิบัติงานได้

หนีไม่พ้น

โดยสรุปแล้วเราก็ต้องหันมาจับเอาพวกจบปริญญาต่าง ๆ มาเป็นโปรแกรม-

เมอร์ แต่ใครจะยอมเป็นแค่โปรแกรมเมอร์ คนที่เรียนจบมาทางสายงานอื่นนั้นเขาก็อยากจะเป็นอะนารีสต์เหมือนกัน เขามีข้ออ้างที่หนักแน่นกว่าเสียอีก คือ เขาเรียนมาทางด้านอื่น มีความเชี่ยวชาญในวิชาชีพของเขา เขาไปเข้าคอร์สทางคอมพิวเตอร์มา 2-3 วิชาแล้ว เขาจึงอยากจะทำหน้าที่ "เอาคอมพิวเตอร์ไปใช้งานในเรื่องที่เขาชำนาญ" มากกว่า ซึ่งก็คือ การออกแบบระบบงานคอมพิวเตอร์นั่นเอง และถ้าเราไปถามคนกลุ่มนี้ว่าใครจะเป็นผู้เขียนโปรแกรมให้ใช้ เขาก็จะตอบว่า "คนที่ศูนย์คอมพิวเตอร์" หรือคนที่เรียนจบด้านคอมพิวเตอร์มานั่นแหละ ในสายตาของคนในสาขาวิชาชีพอื่น ๆ นั้นเขาคิดว่า "คอมพิวเตอร์ไซแอนส์" ก็คือพวกที่เรียนมาเพื่อทำโปรแกรม

ในที่สุดเราก็ต้องวนกลับมาหาคนกลุ่มสุดท้ายคือ "คนที่มีอยู่ในศูนย์คอมพิวเตอร์" เรื่องนี้หนีไม่พ้น คือ คนที่จะเขียนโปรแกรมก็ต้องเป็นคนในศูนย์คอมพิวเตอร์ ซึ่งจากประสบการณ์ที่ผ่านมาเราก็พบว่าจะประกอบไปด้วยคนที่จบปริญญาตรีในสาขาวิชาชีพต่าง ๆ หลายชนิด และรวมทั้งพวกที่เรียนจบทางคอมพิวเตอร์ไซแอนส์และเอ็มไอเอส (MIS) ด้วย คนพวกนี้ต้องเข้ามาฝึกหัดเขียนโปรแกรม, วิเคราะห์ระบบ, และออกแบบระบบ แบบคละ ๆ กันไป แล้วก็ต้องไต่เต้าผ่านการทำหน้าที่ทั้ง 2 ด้านมาโดยตลอด ไม่มีใครที่จะก้าวกระโดดเข้ามาเป็น "อะนารีสต์" อย่างโดดเด่นโดยทันทีได้ หลาย ๆ คนเรียนจบทางคอมพิวเตอร์ไซแอนส์และเอ็มไอเอสและมีประสบการณ์ดีกว่านักเรียนนอกที่เพิ่งจบใหม่ด้วยซ้ำไป แล้วจะให้แต่งตั้งคนเข้ามาใหม่ให้เป็นนายของคนเก่าได้อย่างไร



ผมจึงเห็นว่าความคิดที่จะขอเป็นอะนารีสต์แต่เพียงอย่างเดียว จึงเป็นความคิดที่ผิด ถ้าเขาเป็นคนที่เคยผ่านงานเขียนโปรแกรมและออกแบบระบบมา 10-15 ปีแล้ว และจะขอทำแต่ด้านออกแบบระบบละก็ผมเห็นด้วยอย่างเต็มที่ เพราะเราไม่ควรเอาคนมือดี ๆ อย่างนี้ไปเสียเวลาเขียนโปรแกรมอีกต่อไปแล้ว เขามีความสามารถเพียงพอที่จะไล่ตามเทคโนโลยีโดยไม่ต้องลงมือเขียนโปรแกรมเองอีกต่อไปแล้ว

สรุปส่งท้าย

โดยสรุปแล้ว ปัญหาทั้งหมดเกิดจากการที่นิสิตนักศึกษาส่วนใหญ่ไม่ทราบว่าในชีวิตจริงนั้นเราทำงานกันอย่างไร เด็ก ๆ รุ่นใหม่เหล่านี้ไม่รู้ด้วยซ้ำไปว่าคนที่อยู่ในศูนย์คอมพิวเตอร์ต่าง ๆ นั้นมีกี่คน และมีพื้นฐานกันมาอย่างไร และที่แย่ที่สุดคือ เขาไม่รู้ว่างานจริงนั้นยากกว่าที่คิดหลายพันเท่า เขาคิดว่าเขามีความรู้ความสามารถที่จะออกแบบระบบได้ดี แต่เขาไม่รู้หรอกว่าในงานจริงนั้นต้องการความรู้และประสบการณ์อื่น ๆ อีกมากมาย แต่อย่างไรก็ตามเรื่องนี้จะไปโทษเด็กก็คงไม่ได้ เราในฐานะเป็นผู้ใหญ่ ซึ่งรวมถึงอาจารย์ในมหาวิทยาลัยต่าง ๆ ต้องเป็นผู้ให้ข้อมูลแก่พวกเขา ต้องชี้แนะว่าจะต้องคิดถึงเรื่องอะไรบ้าง และอะไรเป็นไปได้อะไรเป็นไปไม่ได้

ปัญหาของผมในตอนนี้ก็คือ อาจารย์เองนั่นแหละที่ชอบเถียงกันคอเป็นเอ็นเลยว่า ต้องแยกงานออกแบบระบบกับงานเขียนโปรแกรมออกจากกัน ทั้งนี้เพราะอาจารย์จำนวนมากกำลังรับงานภายนอกเข้ามาทำ พวกเขาใช้วิธีออกไปศึกษาและออกแบบระบบเอง แล้วใช้ลูกศิษย์เป็นโปรแกรมเมอร์ เสร็จแล้วก็ไปครอบงำความคิดของนิสิตว่า "เรียนจบออกไปแล้วคุณต้องทำเหมือนผม" ทั้งนี้โดยไม่ดูเลยว่าสภาวะของมหาวิทยาลัยกับหน่วยงานอื่น ๆ นั้นเป็นคนละเรื่องกัน อาจารย์มีลูกศิษย์เป็นลูกน้องได้ แต่เราไม่รู้ว่าจะเอาใครมาเป็นลูกน้องของลูกศิษย์อาจารย์ ลูกศิษย์นั้นอยู่ในมหาวิทยาลัยแค่ 3-4 ปีก็จบแล้ว แต่จะมีใครมายอมเป็นลูกน้องของลูกศิษย์อาจารย์อยู่ตลอดชีพได้ ผมยังไม่รู้ว่าอีกกี่ปีเราจึงจะแก้ปัญหานี้กันได้